High Level Architecture

Interface Specification

Integrated Training Program

High Level Architecture

- **Major functional elements, interfaces, and design rules, pertaining to all DoD simulation applications, and providing a common framework within which specific system architectures can be defined**

- **HLA is the Technical Architecture for DoD Simulations**

Interface Specification

- **Provides a specification of the functional interfaces between federates and the RTI**
  - **65 interfaces in six service groups**

- **Includes:**
  - **Name and Descriptive Text**
  - **Supplied Parameters**
  - **Returned Parameters**
  - **Pre-conditions**
  - **Post-conditions**
  - **Exceptions**
  - **Related Services**

- **Application Programmers Interface in CORBA IDL, Ada'95 and C++**

Six HLA Runtime Infrastructure Service Groups

- **Federation Management**

- **Declaration Management**

- **Object Management**

- **Ownership Management**

- **Time Management**

- **Data Distribution Management**

Federation Management
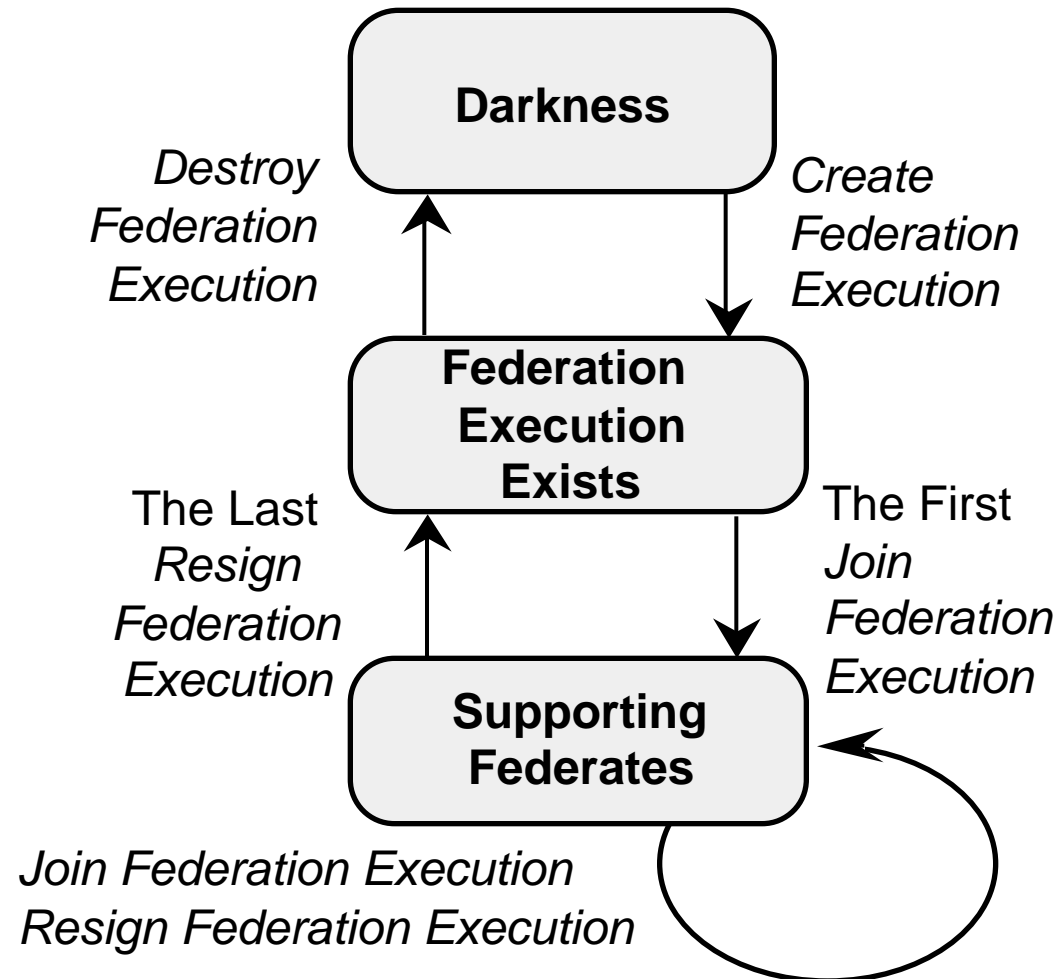
- **Coordinate federation-wide activities throughout the life of a federation execution**
  - **Used by federates to manage a federation execution to meet their needs**
  - **Includes Federation Execution Data (FED)**
    - ◆ **Initializing name space, transportation and ordering defaults and routing space names and dimensions**

- **Interface functions include**
  - **Creation and destruction of a federation execution**
  - **Joining and resigning of a federate**
  - **Coordination of federation saves**
  - **Pausing and resuming a federation execution**

Federation Management

```
                        ┌─────────────┐
                        │  Darkness   │
                        └─────────────┘
  Destroy                ↑        │          Create
  Federation             │        │          Federation
  Execution              │        ▼          Execution
                        ┌─────────────┐
                        │ Federation  │
                        │ Execution   │
                        │   Exists    │
                        └─────────────┘
  The Last               ↑        │          The First
  Resign                 │        │          Join
  Federation             │        ▼          Federation
  Execution                                  Execution
                        ┌─────────────┐
                        │ Supporting  │ ←──┐
                        │ Federates   │    │
                        └─────────────┘    │
                                         ──┘
  Join Federation Execution
  Resign Federation Execution
```

Declaration Management

- **Allow federates to specify the types of data they will send or receive by object class and attribute name and by interaction class from the FOM**

- **Interface functions include specification of:**
  - **Data to be sent:**
    - ◆ **Object classes and attributes and interaction classes that the federate is able to update or send**
  - **Data to be received:**
    - ◆ **Object classes and attributes and interaction classes that the federate is interested to receive**
  - **Controls on data to be sent:**
    - ◆ **Feedback to the federates from the RTI when attribute updates and interactions should be sent given the interest in those by other federates**

Object Management

- **Supports creation, modification, and deletion of objects, their attributes and the interactions they produce**

- **Interface functions include**
  - **Federate requests for IDs**
  - **Registering and discovering objects**
  - **Updating and reflecting object attributes**
  - **Sending and receiving interactions**
  - **Deleting and removing objects**
  - **Changing default transportation and event ordering types**

Time Management

- **Control advancement of federates along with federation time**
  - Coordinated with object management services to support causal behavior across the federation
  - Designed to support federates with different ordering and delivery requirements

- **Interface functions include**
  - Request current values of time
    - Federation time, federate's logical time (LT), lower bound time stamp (LBTS), minimum next event time
  - Set and request lookahead
  - Time advance request, next event and flush queue request, and grant
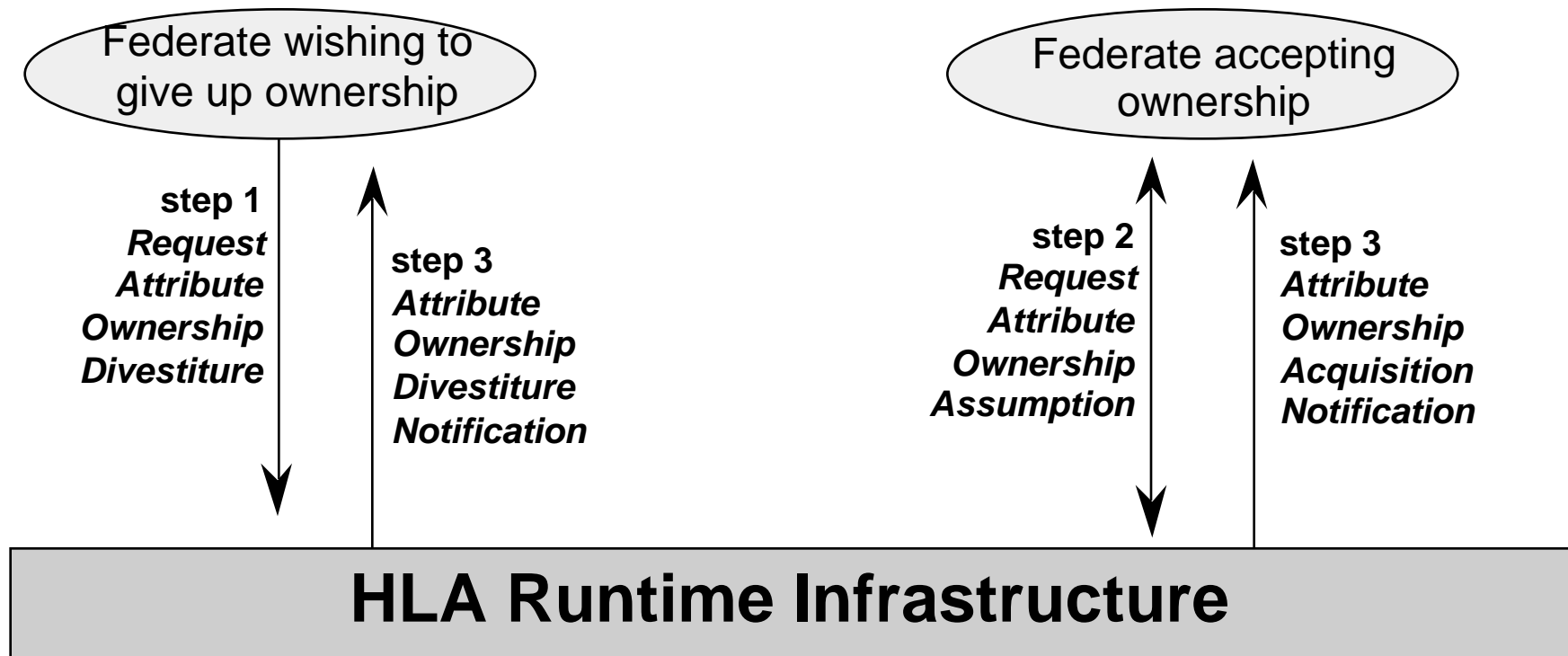
Ownership Management

- **Allow federates to transfer ownership of object attributes**
  - Federates transfer ownership based on federation execution design plans and the RTI arbitrates transactions
  - Offers both 'push' or 'pull' based transactions
    - Acquisition requires current publication and subscription declarations for attribute

- **Interface functions include**
  - Request ownership divestiture and assumption
  - Request ownership acquisition and release
  - Notification of divestiture and acquisition
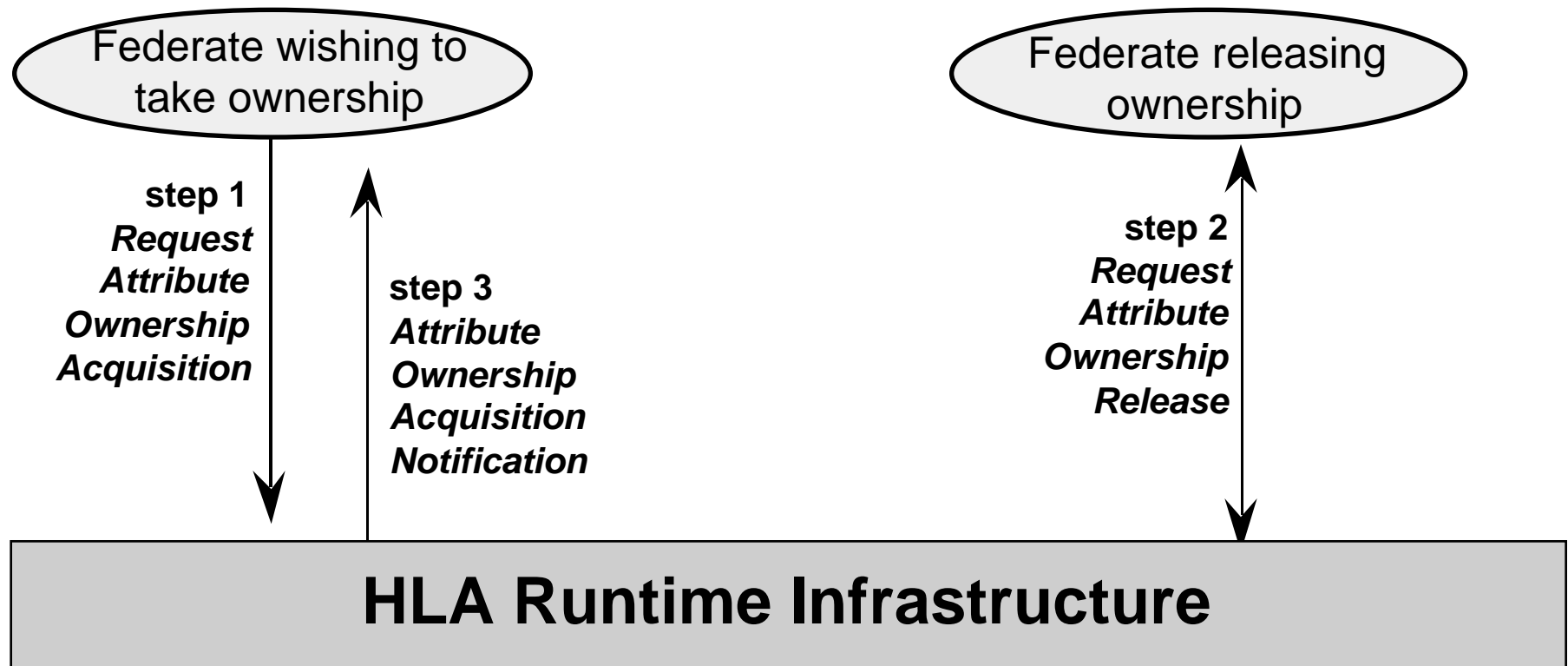 - Query attribute ownership

Divesting Ownership

Federate wishing to give up ownership

Federate accepting ownership

step 1
Request Attribute Ownership Divestiture

step 3
Attribute Ownership Divestiture Notification

step 2
Request Attribute Ownership Assumption

step 3
Attribute Ownership Acquisition Notification

HLA Runtime Infrastructure

Requesting Ownership

Federate wishing to take ownership

Federate releasing ownership

step 1
Request Attribute Ownership Acquisition

step 3
Attribute Ownership Acquisition Notification

step 2
Request Attribute Ownership Release

HLA Runtime Infrastructure

Data Distribution Management

- **Allow federates to specify the distribution conditions for the specific data they send or expect to receive**
    - **RTI uses this information to route data as specified in declaration management services**
    - **Not bound by FOM, data distribution can be managed based on other characteristics of objects important to particular federation execution**
    - **Federation design creates 'routing spaces' for use during runtime; these are specified at federation creation time**

- **Interface functions include**
    - **Create and modify 'update' and 'subscription' regions to bound routing space**
    - **Associate update regions with specific object instances**
    - **Change thresholds for changing regions**

The Role of the Federate in DDM

- **Create Subscription Region**
  - Specify conditions under which they expect to receive the object state data and interactions they specified using declaration management services (Subscribe Object Class Attribute and Subscribe Interaction Class) and

- **Create Update Region**

- **Associate Update Region (with an object instance or interaction)**
  - Specify conditions under which they are providing data (characteristics of object or interaction which map to dimension of routing space fall with region bounds)

- **Modify Region or Associate Update Region**
  - As the state of the objects change, the federate may need to either adjust the bounds on the associated regions or change the association to another region

The Role of the RTI in DDM

- **The routing space, regions, and association data is used by the RTI to distribute data**

- **When an update region and subscription regions of different federates overlap**
  - **The RTI ensures that the attribute updates and interactions associated with that update region are routed to federates with subscription regions which overlap the sender's update region**

- `Change Thresholds`
  - **The RTI provides feedback to federate on the amount of change in extents which will lead to data distribution changes**

Illustration of DDM Services

Two Dimensional Interest Space
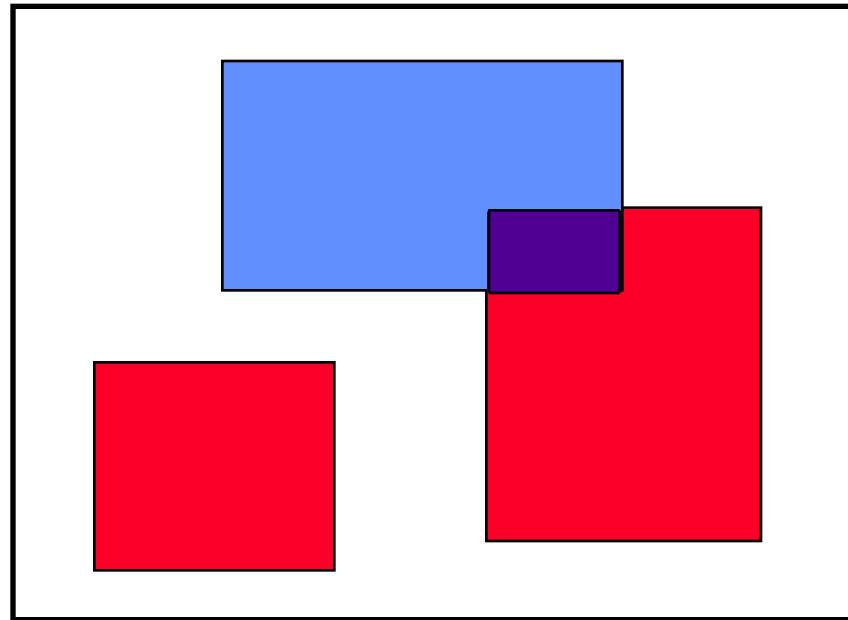
Update Region

Subscription Region

Overlap Region - Published Data Sent to Subscribing Federate

Overview of Federation Execution Life Cycle

Federate
RTI

Create Federation Execution
Join Federation Execution

Establish Initial Data Requirements

Startup

Register Objects
Update Attribute Values
Discover Objects

Normal Federate Execution
Advance Time Request/Grant
Register/Discover Objects
Update/Reflect Attribute Values
Send/Receive Interactions
Delete/Remove Objects

Operations

Resign Federation Execution

Shutdown

Destroy Federation Execution